ISSN: 2321-2152

IJMECE

International Journal of modern electronics and communication engineering

E-Mail
editor.ijmece@gmail.com
editor@ijmece.com

www.ijmece.com

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

SANITIZABLE ACCESS CONTROL SYSTEM FOR SECURE CLOUD STORAGE AGAINST MALICIOUS DATA PUBLISHERS

[1]Pagadala Ganesh Kumar, [2]Challa Varsha, [3]Shivakoti Udayasri

[1]Assistant professor in Department of Information Technology Bhoj Reddy Engineering College for women

[1]ganeshpagadala91@gmail.com

[2,3]UG Scholars in Department of Information Technology Bhoj Reddy Engineering College for women

[2]Ch.Varsha1239@gmail.com ,[3] udayasrishivakoti17@gmail.com

Abstract

Considering that it can significantly reduce the cost of hardware and software resources in one of the best-Known ideas in the information technology sector is cloud computing, which refers to the computer infrastructure. Due to this simplicity of usage, businesses can now successfully data amongst their employees. The simplest solution initially appeared to be to simply save the sharing unencrypted copies of the data kept in.the cloud while keeping access to it secure control. The erroneous presumption that a cloud maintained by a, third party can be trusted is the foundation for this. total confidence that all information will remain confidential. Therefore, encryption is necessary, and the shared Access control must be used with cipher text storage of data. In actuality, though, some of these employees could be rude and unwilling to follow the fundamental guidelines for sharing. Regrettably, The issues caused by poor data suppliers are not currently being addressed. Only The information stored in cloud storage can be decoded by authorized receivers thanks to the literature describes the current form of protection. Although malicious data publishers write data in line with the regulations, the cipher texts may be decoded without their awareness by unauthorized users who have the correct keys or simply by anyone who isn't supposed to. The use of malicious data publishers has a negative impact because it could endanger the company's intellectual property. To this day, the study's goal of determining how to offer a reasonable solution to the problem when there are adversarial data producers in the system. This study examines present a fresh line of inquiry that might address the presence of antagonistic data providers. Through the recommendation of the sanitizable Access Control System (SA-CS), which is designed for secure We solve the aforementioned concerns about malicious data suppliers and cloud storage. The threat model's formal security model, architecture, and methodology are described. It uses the q-Parallel Bilinear Diffie-Hellman Exponent Assumption as its foundation. We think that our strategy has created a completely new

field of study that wasn't there before. had a look at. Our efforts will result in increased adoption of secure cloud storage.

I INTRODUCTION

The emergence of cloud storage technology revolutionized enterprise operations, especially for Small and Medium-sized Enterprises (SMEs), offering low-cost solutions for data sharing. However, trusting the cloud entirely for data security is impractical due to its third-party ownership. Encrypting data before storing it in the cloud is essential to prevent data leakage. Attribute-based Encryption (ABE) has been explored to protect data with access policies, but it falls short when dealing with malicious insiders who intentionally leak sensitive information. Malicious data publishers can encrypt data in a deceptive manner, allowing unauthorized access. The goal is to achieve data privacy even when dealing with malicious insiders who don't adhere to encryption algorithms.

Cloud storage technology has transformed enterprise operations, particularly for Small and Medium-sized Enterprises (SMEs), offering cost-effective solutions for data sharing. However, fully trusting the cloud for data security is unrealistic due to its third-party ownership. Encrypting data before storing it in the cloud is crucial to prevent data leakage. Attribute-based Encryption (ABE) has been proposed to safeguard data with access policies, but it falls short when dealing with malicious insiders who intentionally leak sensitive information. Malicious data publishers can encrypt data deceptively, enabling unauthorized access. The aim is to ensure data privacy even when confronting malicious insiders who disregard encryption protocols.

II RELATED WORK

Access control is able to guarantee data security in cloud storage systems. This has attracted much attention from academia and industry. IBM developed the capability-based model and systematic approaches to improve access control in the cloud services. Cryptographic primitives have been proposed for enabling access control on encrypted storage, such as broadcast encryption, proxy re-encryption, role-based encryption and attribute-based encryption. For the reason of security, scalability and flexibility, ABE has been regarded as one of the most suitable technologies for enabling access control. Users whose attributes satisfying the

access policy are able to access the plain data. ABE is mainly classified into two complementary forms, key-policy ABE and ciphertext-policy ABE. In CP-ABE, attributes are used to describe the user's attributes and access policies over these attributes are attached to the encrypted data. Due to its flexibility and expressiveness, CP-ABE has more applications in cloud storage access control.

•Sanitizable Signatures

Sanitizable signatures (SS's) are proposed by Ateniese et al to allow controlling modifications of signed messages without invalidating the signature. SS is a variant of digital signatures where a designated party (the sanitizer) can update admissible parts of a signed message. Brzuska et al introduced most of security notions in SS's. Fehr and Fischlin proposed sanitizable signcryption to hide the message-signature pair from the sanitizer. Many SS schemes have been proposed to satisfy different properties. SS provides the foundation to the concept of sanitization in encryption.

•Access Control Encryption

Access Control Encryption (ACE)was introduced to provide fine-grained access control. ACE gives different rights to different users not only in terms of which messages they are allowed to receive, but also which messages they are allowed to send. Here, the important property of Sanitization is included. ACE can prevent corrupted senders from sending information to corrupted receivers. In ACE, the sanitizer uses its sanitizer key from the authority to execute a specific randomized algorithm on the incoming ciphertext and thereafter passes the result to a database server or the receivers. By sanitizing, ACE ensures that no matter what the corrupted sender sends, what the receiver receives looks like a random encryption of a random message. In our SACS, the sanitizing operation does not need a sanitizer key from the authority. Only the valid receiver, who is assigned a valid private key by the authority, can recover the message.

III EXISTING SYSTEM

The existing system make use of the notion of Attribute-based Encryption(ABE) to enable such an unauthorized prevention by protecting the data with an appropriate access policy. Anyone who is equipped with a valid decryption key satisfying the access policy will be able to decrypt the data correctly. It implies that data will be stored as a cipher text instead of a plaintext in the cloud. This kind of protection only considers data privacy when data publishers are honest and follow the encryption algorithm.

Unfortunately, in practice, some of these employees may be malicious and they intentionally attempt to leak the contents of those data to unauthorized recipients, such as competing business corporations. These malicious employees may even would like to

publish some sensitive contents and store them in the cloud, but also enable other unauthorized users to retrieve it – hence constituting a malicious data publisher. It is unfortunate that the approach based on ABE is not sound due to malicious data publishers, who encrypt data in a pernicious way.

Problems in Existing System:

- The existing protection in the literature has been explored to allow only legitimate recipients to decrypt the contents stored in the cloud storage, but unfortunately, no existing work deals with issues raised due to the presence of malicious data publishers.
- It may damage intellectual properties from the corporations.
- Existing system is not completely secured.
- In the existing system, the malicious data publishers will construct ciphertexts containing copyrighted materials that seemingly adhere the required security access policy by the organization. Nevertheless, the supposedly encrypted file can be decrypted by anyone illegally without any valid decryption key.

IV PROBLEM STATEMENT

The problem statement revolves around designing a sanitizable access control system tailored for secure cloud storage, specifically focusing on mitigating threats posed by malicious data publishers. This system needs to ensure that unauthorized or maliciously modified data is not distributed or accessed, while still allowing authorized users to interact with the stored information. The main challenge lies in developing a robust mechanism that can identify and isolate potentially harmful data while maintaining efficient and legitimate access for authorized users within a cloud storage environment decrypted by anyone illegally without any valid decryption key.

V PROPOSED SYSTEM

One main goal is to achieve data privacy when data publishers are malicious and they do not follow the encryption algorithms accordingly. We aim to propose a very practical notion called Sanitizable. Access Control System, or simply SACS, which is designed for the cloud storage to resist against malicious data publishers. SACS enables a flexible access control for both data publishers and data receivers.

SACS is equipped with sanitizing capability, which prevents malicious data publishers from generating cipher texts that will be decrypt able without any valid private keys. Although the malicious data publishers can maliciously generate cipher texts which can be decrypted by anyone, the sanitizer will transform these cipher texts into new cipher texts which will only be decrypt able by valid private key holders.

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

We prevent our architecture as well as our scheme to achieve the above concept to build SACS. Furthermore, we also present an implementation of SACS. We call the entity who sends the cipher data as the data publisher and the entity who recovers the plain data as the receiver. We still consider cloud as the data storage platform.

Advantages of Proposed System:

•SACS aims to provide a flexible access control in terms of both data receivers and data publishers.

•It enables only valid data receivers, who bold private keys from the authority, to access the plain data.

•With ciphertext sanitizing, SACS prevents malicious data publishers from issuing information that can retrieve decryption key without valid private key generated from the trusted center. e.g. encryption key, such that any invalid receivers cannot access plain data even if they have encryption key.

•SACS allows to sanitize the cipher data. The sanitization is to prevent any data publisher from malicious behaviors, which incurs invalid access to plain data. SACS makes an enhancement

•to data privacy in terms of publishers and provides secure data storage against malicious data publishers.

•SACS ensures stronger access control over encrypted data. It guarantees that only valid receivers can access the plain data while any receiver, even if he/she has an encryption key from the malicious data publisher, cannot decrypt the sanitized cipher data correctly.

VI IMPLEMENTATION

Authority:

The authority manages and maintains the whole system. In SACS, we regard the authority as a trusted entity who holds the master secret key. The authority issues a unique private key to each receiver who registers into this system. Without loss of generality, we assume that the authority neither colludes with any other entities nor is compromised.

Data Publisher:

The data publisher owns the plain data. He/she encrypts its plain data with an encryption key (e.g., K) and sets an access policy to deal with the encryption key. Then the data publisher sends the encrypted data (or cipher data) to the sanitizer. Actually, the publisher relies on this access policy to conduct data access control. Publishers are either honest or malicious. Both honest and malicious publishers execute the encryption operation on the plain data, but a malicious publisher might have extra behaviors, such as distributing the encryption key to some non-registered receivers. This incurs a failure of

ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024

access control since some receivers can access the data without valid private key.

Sanitizer:

The sanitizer is introduced to transform the original cipher data into the sanitized cipher data. Once getting cipher data from the data publisher, the sanitizer is instructed to do some specific processing on these cipher data. The processing includes two parts. One is to check whether the cipher data is under the claimed access policy and the other is to sanitize the cipher data with its encryption key K0. Then the sanitizer sends the sanitized cipher data to the cloud server for storage. Such a sanitization operation on the cipher data is to prevent malicious publishers and invalid data access. The sanitizer is an honest party, which means it just executes the sanitization following the sanitizing algorithm but no malicious operations, such as replacing/modifying the cipher data. The sanitizer learns nothing about the plain data.

Receiver:

The receiver wants to access the plain data. He/she can freely download the cipher data that he/she is interested in from the cloud server. Prior to accessing the data, the receiver must register into the system and ask for a private key from the authority. When the registered receiver owns conditions satisfying the access policy, it is valid. Only valid receivers can access the plain data from the data publisher. Receivers will share neither their private keys nor the decrypted plain data with other entities. Here, we note that each receiver is unique.

Cloud Server:

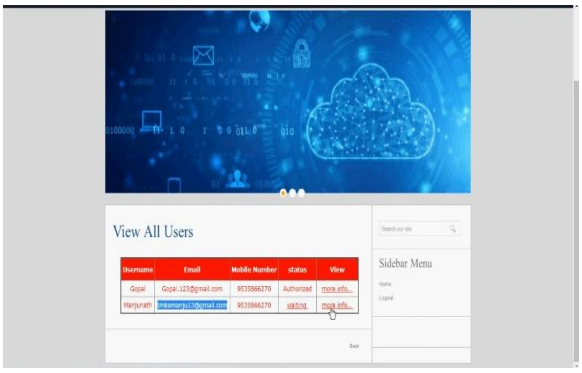
The cloud server provides a platform for cipher data storage. The cipher data stored in the cloud server can be acquired by any receivers. The cloud server just receives cipher data from the sanitizer and sends the cipher data to the receiver, while executes no computation operation. The cloud server will behave maliciously, e.g., delete the cipher data. Whether the cloud server is curious or not gives no effect on the security of SACS.

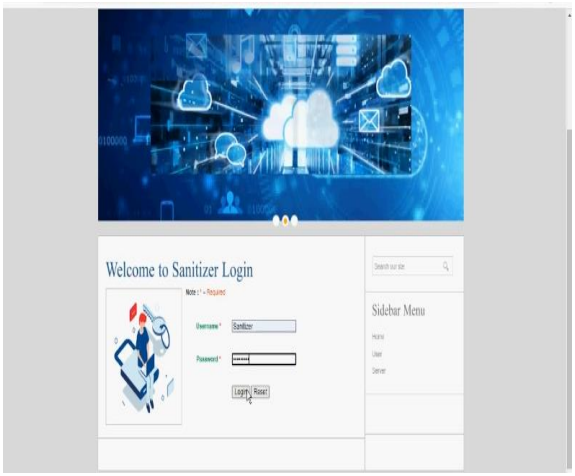
VII RESULTS

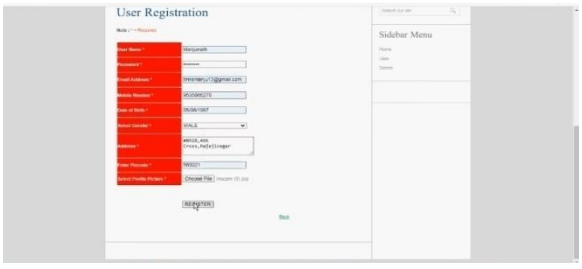
ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024



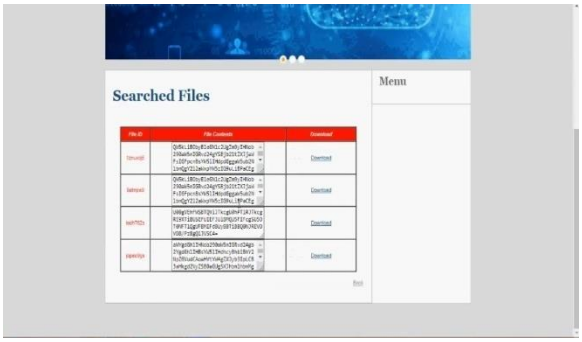


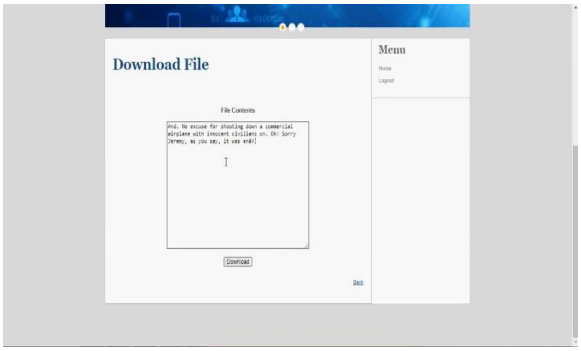


ISSN2321-2152

www.ijmece .com

Vol 12, Issue.2 April 2024





VIII CONCLUSION

In conclusion, a sanitizable access control system plays a crucial role in fortifying secure cloud storage against threats posed by malicious data publishers. By implementing robust access controls, encryption methods, data validation, behavioral analysis, and potentially AI-driven solutions, these systems aim to mitigate the risks of unauthorized access, data manipulation, or infiltration by malicious entities.

However, the landscape of cyber security is continually evolving, demanding ongoing enhancements to these systems. Innovation in technologies and methodologies is essential to counter emerging threats and adapt to evolving attack vectors. Collaboration among researchers, industry experts, and regulatory bodies remains pivotal to strengthen these systems and ensure the integrity, confidentiality, and availability of data stored in cloud environments

REFERENCES

[1] V. Goyal, O. Pandey, A. Sahai, and B. Waters, “Attribute-based encryption

for fine-grained access control of encrypted data,” in Proceedings of

the 13th ACM Conference on Computer and Communications Security,

CCS 2006, 2006, pp. 89–98.

[2] J. Bethencourt, A. Sahai, and B. Waters, "Ciphertext-policy attributebased

encryption," in 2007 IEEE Symposium on Security and Privacy

(S&P 2007), 2007, pp. 321–334.

[3] B. Waters, "Ciphertext-policy attribute-based encryption: An expressive,

efficient, and provably secure realization," in PKC 2011, ser. LNCS, vol.

6571, 2011, pp. 53–70.

[4] S. Berger, S. Garion, Y. Moatti, D. Naor, D. Pendarakis, A. Shulman-

Peleg, J. R. Rao, E. Valdez, and Y. Weinsberg, "Security intelligence

for cloud management infrastructures," IBM Journal of Research and

Development, vol. 60, no. 4, pp. 11:1–11:13, 2016.

[5] "Secure access control for cloud storage," https://www.research.ibm.

com/haifa/projects/storage/cloudstorage/secure access.shtml.

[6] D. Boneh, C. Gentry, and B. Waters, "Collusion resistant broadcast

encryption with short ciphertexts and private keys," in CRYPTO 2005,

ser. LNCS, vol. 3621, 2005, pp. 258–275.

[7] G. Ateniese, K. Fu, M. Green, and S. Hohenberger, "Improved proxy

re-encryption schemes with applications to secure distributed storage,"

ACM Trans. Inf. Syst. Secur., vol. 9, no. 1, pp. 1–30, 2006.

[8] L. Zhou, V. Varadharajan, and M. Hitchens, "Achieving secure rolebased

access control on encrypted data in cloud storage," IEEE Trans.

Information Forensics and Security, vol. 8, no. 12, pp. 1947–1960, 2013.